Worksheet 3 Bubble sort and insertion sort

Task 1

1. Use the girls' name cards for Exercise 1.
   Place the cards in ascending order of popularity, i.e.
   Sophie, Lily, Jessica, Isabella, Ava, Poppy, Emily, Isla, Olivia, Amelia
   Perform a manual bubble sort to get the cards in alphabetical order.

   What are the last 2 names after the first pass?

   What are the first two names after the second pass?

   What are the third and fourth names after the third pass?

   What are the seventh and eighth names after the fourth pass?

   What are the fourth and fifth names after the fifth pass?

   What are the first two names after the sixth pass?

   Which names are out of sequence after the seventh pass?

   Are any names out of sequence after the eighth pass?

   How many passes were needed to sort the cards?

2. Complete the bubble sort algorithm given below.

```
for i = 0 to n - 2
  for j = 0 to (n – i - 2)

      if




      endif

  next j
next i
```

Task 2

3. Place the name cards in ascending order of popularity:

   Sophie, Lily, Jessica, Isabella, Ava, Poppy, Emily, Isla, Olivia, Amelia

   Perform a manual insertion sort on the cards to put them into alphabetical order.

   What sequence are the first four cards in after 4 moves have been made?

4. The following numbers are to be sorted into ascending order using an insertion sort:

   15, 73, 29, 66, 35, 11, 43, 21

   (a) Show the sequence of the numbers after each pass through the insertion sort algorithm

```
aList = [15, 73, 29, 66, 35, 11, 43, 21]
#assume first element of array is aList[0]
for j = 1 to len(aList) - 1
   nextNum = aList[j]
   i = j – 1
   while i >= 0 and aList[i] > nextNum
      aList[i + 1] = aList[i]
         i = i - 1
   endwhile
   aList[i + 1] = nextNum
next j
```

15	73	29	66	35	11	43	21

(b) How many passes are made through the data for a dataset of n numbers?

Extension task

5. Write an algorithm which will compare the length of time to sort a list of n random numbers using a bubble sort and an insertion sort.

   If you have time, write the program in a language of your choice and experiment with different values of n. Be prepared to wait a long time if you choose a number greater than 10,000!

   Which is fastest for 10 numbers?
   Which is fastest for 100 numbers? 1000 numbers?